Update on Wiki

- 更新了本周周报
- 将周报全放在了一个新的界面
- 更新了技能书
- 更新了会议记录
- 将会议记录全放在了一个新的界面

团队训练

2019 Multi-University Training Contest 1

每周推荐

王兴罡 推荐一道加深理解线性基的题 hdu 6579 题解见周报团队训练
傅云濠 推荐一道练习数学推导的反演题（好像可以不用反演），题解详见周报团队训练

个人训练

傅云濠

专题

计算几何——半平面交
帮忙更新“ 树套树 ”

比赛

个人赛没打

题目

- 训练赛补题

- BZOJ3110 树套树
- P4196 半平面交
- POJ2451 半平面交

王兴罡

专题

树套树，详情见wiki

比赛

无

题目

hdu 6579

题意

给了一个序列，要求实现两种操作

1. 给定 l,r 求 $a[l..r]$ 种选出其中的一些值的最大异或和
2. 在序列的后面加一个 x □

题解

开始想到线段树套线性基，发现时间和空间都爆了。后发现我们可以记录 $a[1..i]$ 的线性基，添加时候则从高位到低位，尽量用当前的基去替换之前的基，这样能使所有的基离r更近。查询的时候只用位置大于 l 的基。

黄旭民

专题

复习了多项式，整理了多项式部分模板□NTT□多项式求逆，多项式积分，多项式求导，多项式求ln□

比赛

无

题目

hdu6586

题意

给定一个只包含小写字母的字符串，要求选出一个长度为K的子序列，满足字典序最小，同时子序列中每个字母的出现次数都在一个各自给定的范围$[L,R]$内

题解

贪心。 当前位置能选字典序小的就选字典序小的。 对于子序的第i个位置，从a开始枚举到z尝试将当前位置后第一个该字符放入，然后看看后面剩下的子串能不能至少满足能构成长度为K的合法子串。 具体查看如下： 1、剩下能用字符能否至少构成长度为K 2□剩下每个字符数量能否至少达到下界L_i□ 这个下界即要查看每个字符的剩余字符数是否足够，还要查看子序列剩余字符数能否提供所有的字符达到下界。

然后就完了。【很不明白比赛是怎么没调出来，，】

```
#include<algorithm>
#include<iostream>
#include<cstdlib>
#include<cstring>
#include<cstdio>
#include<vector>
#include<queue>
#include<cmath>
#include<map>
#include<set>
#define LL long long int
#define REP(i,n) for (int i = 1; i <= (n); i++)
#define Redge(u) for (int k = h[u],to; k; k = ed[k].nxt)
#define cls(s,v) memset(s,v,sizeof(s))
#define mp(a,b) make_pair<int,int>(a,b)
#define cp pair<int,int>
using namespace std;
const int maxn = 100005,maxm = 100005,INF = 0x3f3f3f3f;
inline int read(){
    int out = 0,flag = 1; char c = getchar();
    while (c < 48 || c > 57){if (c == '-') flag = 0; c = getchar();}
```

```
    while (c >= 48 && c <= 57){out = (out << 1) + (out << 3) + c - 48; c =
getchar();}
    return flag ? out : -out;
}
vector<int> pos[26];
int pi[26],K,n;
int sum[maxn][26];
int cnt[maxn],L[maxn],R[maxn];
char s[maxn],ans[maxn];
void init(){
    for (int i = 0; i < 26; i++){
        cnt[i] = 0; pos[i].clear(); pi[i] = 0;
    }
}
int main(){
    while (~scanf("%s",s + 1)){
        init();
        K = read(); n = strlen(s + 1);
        for (int i = 0; i < 26; i++) L[i] = read(),R[i] = read();
        for (int i = 1; i <= n; i++){
            int u = s[i] - 'a';
            pos[u].push_back(i);
            for (int j = 0; j < 26; j++) sum[i][j] = sum[i - 1][j];
            sum[i][u]++;
        }
        //cout << sum[2][0] << endl;
        int u = 1,tag = true;
        for (int i = 1; i <= K; i++){
            tag = false;
            for (int j = 0; j < 26; j++){
                //printf("[%d,%d]\n",i,j);
                if (cnt[j] >= R[j]) continue;
                while (pi[j] < pos[j].size() && pos[j][pi[j]] < u) pi[j]++;
                if (pi[j] >= pos[j].size()) continue;
                int tmp = 0;
                //puts("LXT");
                for (int k = 0; k < 26; k++) tmp += min(R[k] -
cnt[k],sum[n][k] - sum[pos[j][pi[j]] - 1][k]);
                if (tmp < K - i + 1) continue;
                //puts("LXT");
                int flag = true; tmp = 0;
                for (int k = 0; k < 26; k++){
                    if (sum[n][k] - sum[pos[j][pi[j]] - 1][k] < L[k] -
cnt[k]) flag = false;
                    if (k != j){
                        if (L[k] > cnt[k]) tmp += L[k] - cnt[k];
                    }
                }
                if (tmp > K - i) continue;
```

```
                //puts("LXT");
                if (!flag) continue;
                cnt[j]++;
                ans[i] = j + 'a';
                u = pos[j][pi[j]++] + 1;
                tag = true;
                //puts("pick");
                break;
            }
            if (!tag) {puts("-1"); break;}
        }
        if (tag){
            for (int i = 1; i <= K; i++) putchar(ans[i]);
            puts("");
        }
    }
    return 0;
}
```